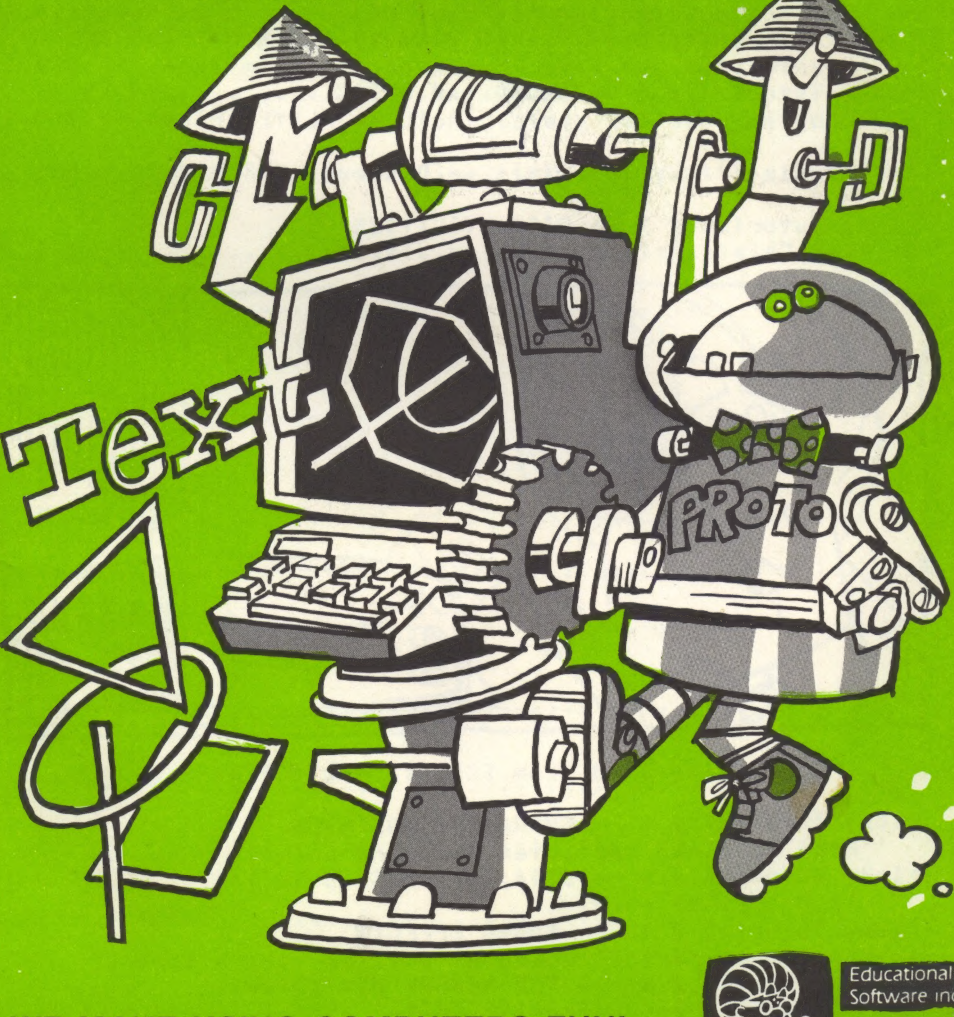
PROTO'S GRAPHICS MACHINE

FOR ATARI® COMPUTERS

BY GEOFF CARAS



WE MAKE USING COMPUTERS FUN!

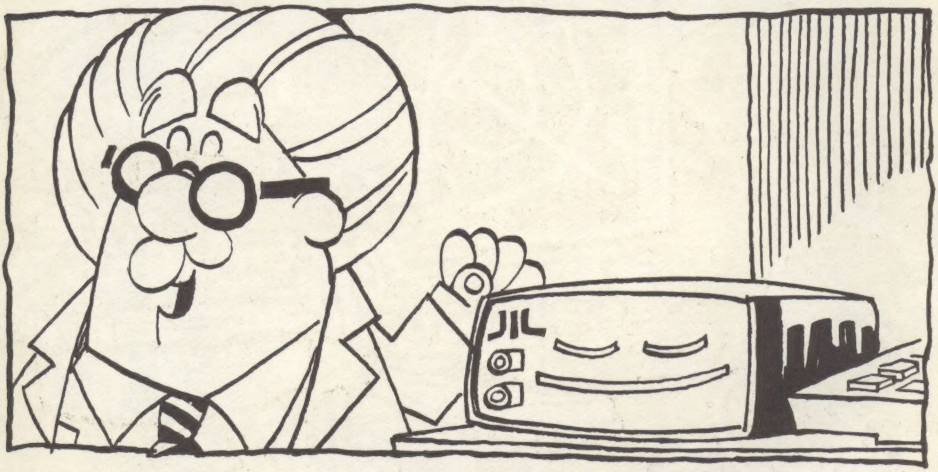
Educational Software inc.
SOQUEL, CA

ALSO FROM EDUCATIONAL SOFTWARE

Professor Von Chip's TRICKY TUTORIALS

Here, at last, are the means for knowing, using, and understanding the great power inherent in the ATARI 400/800. Von Chip's TRICKY TUTORIALS take the mystery out of programming by offering several coherent lessons such as:

Tricky Tutorial # 1 -- DISPLAY LISTS A lesson on mixed graphic screens	-- $19.95
Tricky Tutorial # 2 -- HORIZONTAL/VERTICAL SCROLLING Scroll up, down, sideways, or even diagonally	-- $19.95
Tricky Tutorial # 3 -- PAGE FLIPPING Change screens instantly!	-- $19.95
Tricky Tutorial # 4 -- BASICS OF ANIMATION Beginning animation the easy way.	-- $19.95
Tricky Tutorial # 5 -- PLAYER MISSILE GRAPHICS Create your own video games and animate them.	-- $29.95
Tricky Tutorial # 6 -- SOUND and MUSIC Program your own music, includes PLAYER PIANO	-- $19.95
The 1st Six Tricky Tutorials -- Discount Price	**-- $99.95**
Tricky Tutorial # 7 -- DISK UTILITIES Von Chip's tricks on using disk drives.	-- $29.95



Coming Soon from EDUCATIONAL SOFTWARE

Tricky Tutorial # 8 -- CHARACTER GRAPHICS
Make your own character sets and animate them.

Tricky Tutorial # 9 -- GTIA, GRAPHICS 9 TO 11
16 Colors at once!

Tricky Tutorial # 10 -- SOUND EFFECTS
Find that perfect sound effect for your program.

Tricky Tutorial # 11 -- THE MEMORY MAP TUTORIAL
30 lessons demonstrating useful memory locations.

CONTENTS

INTRODUCTION

The Graphics Machine is a software package that turns your ATARI 800 into a powerful graphics terminal. With the Graphics Machine, you can create graphic designs on your television screen, and save these designs in disk files. To see some examples of what the Graphics Machine can do, remove the BASIC Cartridge from your computer. You see the Graphics Machine prompt.

```
>
```

After the prompt, type:

```
FILER 0
```

Press the RETURN key. FILER is the Graphics Machine command that retrieves command files from the disk (it is explained in the DISK FILES section of this manual). The Graphics Machine asks you for the name of the file to retrieve. You see:

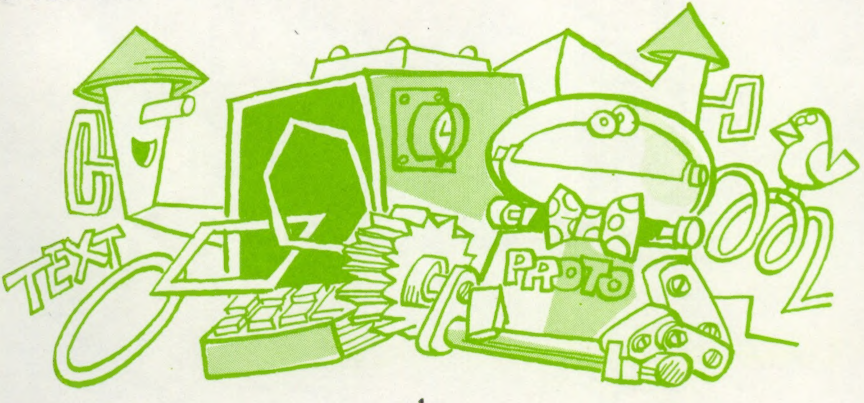
```
filename==>?
```

Type:

```
DEMOLOOP
```

The demonstration examples you see are samples of the kinds of images that you can create on your television screen with the Graphics Machine. This manual explains how to use Graphics Machine commands to draw geometric figures, add colors, create your own data and command files on disks, and retrieve files from disks. Most commands have demonstration examples on the master disk that show you the images the command can produce.

The Graphics Machine is simple to use, but it will be even simpler if you take a few minutes to read about its commands and command structures before you start creating your own graphic designs. In order to read this information accurately, you will need to know the conventions used to describe it.



Conventions Used in this Manual

* Graphics Machine commands are capitalized. For example, BOX, FILER, and FULLSCREEN are Graphics Machine commands. Sample commands are separated from text and appear just as they do on your monitor screen.

* Command parameters, specific instructions which accompany commands, are always numbers. It often makes more sense to refer to a command parameter with a descriptive word. All names given to command parameters in this manual are lowercase. For example, you may see:

```
BOX x1,y1,x2,y2
```

In this case, x1,y1,x2,y2 are names given to the command parameters, Remember, these names represent numbers; whenever you enter a command you type the appropriate numbers NOT the descriptive names.

* Words you are asked to type always FOLLOW a prompt. Type these words exactly as you see them.

The Graphics Mode Eight Screen

The Graphics Machine plots figures on the standard ATARI Graphics Mode Eight Screen. In Graphics Mode eight, the screen is defined by a Cartesian coordinate system, with 320 (0 to 319) points on the horizontal axis and 192 (0 to 191) points on the vertical axis. Figure 1 shows you the dimensions of the Graphics Mode eight screen.

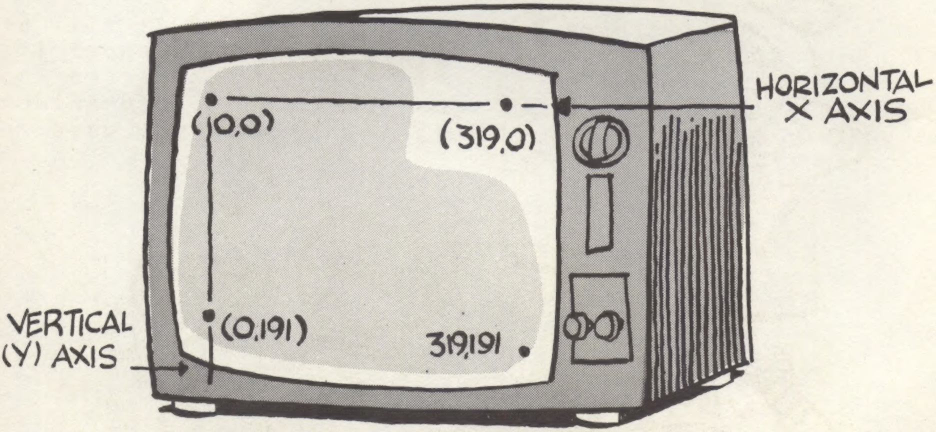


Figure 1: The Graphics Mode Eight Screen

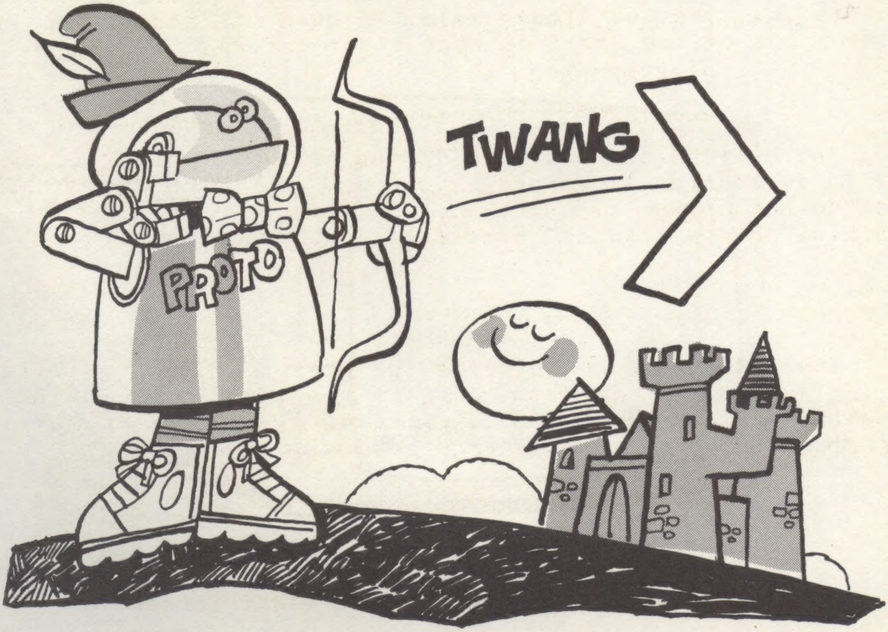
You specify a certain point on the screen using the format (x,y), where x and y are horizontal and vertical coordinates of that point respectively. For example, the origin (0,0) is the upper left-hand corner of the screen, and the point (319,191) is the lower right-hand corner of the screen. The point (50,100) is the intersection of two lines, one passing through the point 50 on the horizontal axis, and the other intersecting it at the point 100 on the vertical axis.

The bottom four text-sized lines are called the "command winaow". You see the commands that you type in this window.

Basic Command Structure

The "greater than" symbol (>) is the Graphics Machine prompt. Whenever you see this prompt, the Graphics Machine is waiting for you to type a command.

A command consists of a name followed by one to six parameters, each separated by a comma. Parameters are numbers that tell the Graphics Machine specific information about a command. For example, some parameters tell the Graphics Machine where to locate a figure, others may tell it the dimensions of the figure. The exact meaning of a parameter depends upon the command, and every command must be accompanied by at least one parameter. Many commands, however, do not need any specific information; these commands have only one parameter: the number zero (0).



HELPFUL COMMANDS

HELP

If you type the word "HELP" anywhere on the command line, the Graphics Machine displays a summary of its commands and their syntax on the screen. Even if you were to type a line such as:

```
> SJIUJ(9UJLJLPO**HELP**(UTYYOUJLO)UJUTG
```

the Graphics Machine would respond with the command summary. If you make an error in typing a command three times consecutively, the Graphics Machine will remind you to:

```
type HELP for Command Summary
```

DISKERRORS

The DISKERRORS command tells the Graphics Machine to print descriptive error messages whenever you make a mistake in entering a command.

Command Syntax: DISKERRORS

```
> DISKERRORS 0
```

You must have the disk with the help1 and help2 files on it in disk drive 1, in order to use DISKERRORS command. If you do not enter the DISKERRORS command, the Graphics Machine prints standard numerical error codes.

ERROROFF

The ERROROFF command tells the Graphics Machine to stop printing long error messages; it disables the DISKERRORS command. After the ERROROFF command, the Graphics Machine prints standard numerical error codes.

Command Syntax: ERROROFF

```
> ERROROFF 0
```

LINE

The LINE command draws straight lines on the screen. You tell the Graphics Machine where to begin and end the line by the input parameters: x1,y1,x2,y2. (x1,y1) are the coordinates of one end point, (x2,y2) are the coordinates of the other.

Command Syntax: LINE

```
> LINE x1,y1,x2,y2
```

x1-- The x-coordinate of the first endpoint.
y1-- The y-coordinate of the first endpoint.
x2-- The x-coordinate of the second endpoint.
y2-- Tje y-coordinate of the second endpoint.

To see some sample LINE commands, type:

```
> FILER 0
filename==>? LINE.DEM
```

The LINE demo shows you the lines created by the following LINE commands:

```
> LINE 0,0,319,0
> LINE 319,0,319,191
> LINE 319,191,0,191
> LINE 0,191,0,0
> LINE 0,0,319,191
> LINE 0,191,319,0
```

Notice that these commands draw the border of the Graphics Mode Eight Screen. Normally, you cannot see the lines or the parts of the lines that are behind the command window. Even though you do not see figures behind the command window, the Graphics Machine plots them, just as it does figures you do see. The command that lets you view the full graphics screen (FULLSCREEN) is described later.

You can use the LINE command to plot a single point. Type the coordinates of that point twice, as both the first and the second endpoint.

BOX

The BOX command will draw any rectangular shape. You specify the dimensions by the input parameters: x1,y1,x2,y2. (x1,y1) are the coordinates of the upper left-hand corner of the box.

Command Syntax: BOX

```
> Box x1,y1,x2,y2
```

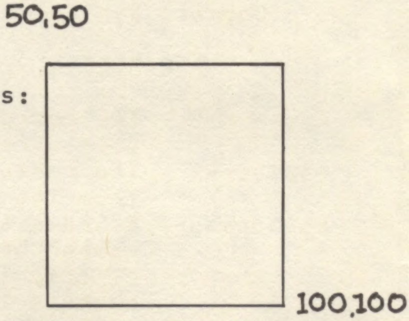
x1-- The x-coordinates of the upper left-hand corner.
y1-- The y-coordinates of the upper left-hand corner.
x2-- The x-coordinates of the lower right-hand corner.
y2-- The y-coordinates of the lower right-hand corner.

To see some sample BOX commands, type the following:

```
> FILER 0
filename==>? BOX.DEM
```

The BOX demo draws the following boxes:

```
> BOX 50,50,100,100
> BOX 0,0,150,150
> BOX 83,75,300,155
> BOX 75,15,85,25
> BOX 120,10,130,100
```

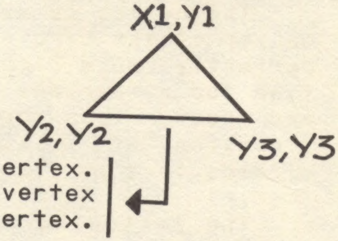


TRIANGLE

The TRIANGLE command draws lines connecting three points on the screen. The command parameters are the triangle's three vertices.

Command Syntax: TRIANGLE

```
> TRIANGLE x1,y1,x2,y2,x3,y3
```



x1,y1-- The coordinates of the first vertex.
x2,y2-- The coordinates of the second vertex
x3,y3-- The coordinates of the third vertex.

To see sample TRIANGLE commands, type:

```
> FILER 0
filename==>? TRIANGLE. DEM
```

The TRIANGLE demo shows you the following triangles:

```
> TRIANGLE 300,0,110,150,99
> TRIANGLE 110,115,150,105,300,155
> TRIANGLE 250,50,155,103,275,140
> TRIANGLE 90,100,100,120,60,110
```

CIRCLE

The CIRCLE command draws circles and ellipses on the screen. You must tell the Graphics Machine: the coordinates of the center of the circle, its radius, and its roundness.

Command Syntax: CIRCLE

```
> CIRCLE x,y,radius,roundness
```

x,y-- The coordinates of the center point of your circle.

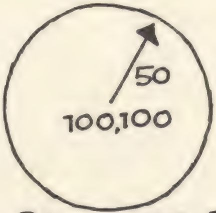
radius-- The radius, measured in horizontal units from the center.

roundness-- Roundness is measured relative to the number one. To plot a perfectly round circle, type the number one. Numbers less than one and greater than zero elongate in the vertical direction. Numbers greater than one elongate in the horizontal direction.

The radius is measured in horizontal units from the center of the circle. Roundness is measured relative to a perfectly round circle. To draw a perfectly round circle, type the number one as the roundness. Numbers less than one and greater than zero elongate your circles in the vertical direction, and numbers greater than one elongate your circles in a horizontal direction.

The following commands all draw circles and ellipses with a center of (100,100) and a radius of 50. Only the different degress of roundness affect the way they look on the screen.

```
> CIRCLE 100,100,50,8
> CIRCLE 100,100,50,1
> CIRCLE 100,100,50,2
> CIRCLE 100,100,50,5
```



ROUNDNESS=1

To see these circles, type:

```
> FILER 0
filename==> CIRCLE.DEM
```

POLYGON

The POLYGON command lets you draw many-sided figures at different angles of rotation.

Command Syntax: POLYGON

```
> POLYGON x,y,radius,number of sides,angle of rotation
```

```
x,y--                 The coordinates of the center of
                      the POLYGON.
radius--              The radius measured in horizontal
                      units from the center.
number of sides--     The number of sides of the POLYGON
angle of rotation--   The measure in degrees of the
                      rotation of the starting point of
                      the plot. An angle of zero degrees
                      starts the plot at the point
                      (x1 + radius,y1)
```

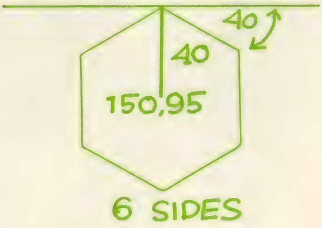
In order to plot a POLYGON, the Graphics Machine must know: the coordinates of the center , the radius in horizontal units from the center, the number of sides, and the angle of rotation.

Notice that these parameters are similar to the parameters for the CIRCLE command. In fact, a POLYGON with more than fifteen sides appears as a circle on your screen.

The center and the radius are exactly the same as with the CIRCLE command, number or sides is self-explanatory, but angle of rotation may seem a bit more confusing. The best way to understand the angle of rotation is to actually try several POLYGON commands. Look at the POLYGON demo which plots several polygons-the only differences between these polygons are their angles of rotation.

Type:

```
> FILER 0
filename==> POLYGON.DEM
```



The POLYGON demo plots these polygons:

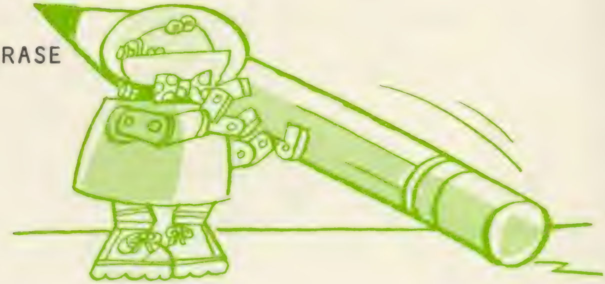
```
> POLYGON 150,95,50,6,0
> POLYGON 150,95,40,6,20
> POLYGON 150,95,40,6,40
> POLYGON 150,95,40,6,60
```

ERASE

To erase a figure you have plotted with the Graphics Machine, you simply plot the figure again--this time in the same color as the background. The ERASE command tells the Graphics Machine to plot all subsequent commands in the background color.

Command Syntax: ERASE

```
> ERASE 0
```



For instance, the following sequence of commands draws a circle and then erases that circle.

```
> CIRCLE 50,50,45,1
> ERASE 0
> CIRCLE 50,50,45,1
```

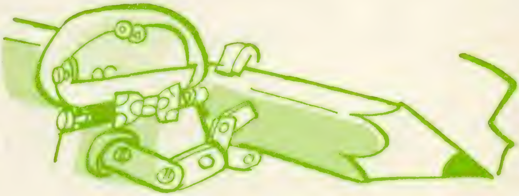
Whenever the Graphics Machine is in ERASE mode, you see an inverse-video prompt (<). The Graphics Machine remains in ERASE mode until you tell it to return to normal DRAW mode; do this with the DRAW command.

DRAW

The DRAW command instructs the Graphics Machine to return to DRAW mode when you are in ERASE mode. After the DRAW command, the Graphics Machine plots commands in white on the black background.

Command Syntax: DRAW

```
> DRAW 0
```



It is not an error to type the DRAW command when you are already in DRAW mode. In fact, it is a good idea to put a DRAW command at the beginning of all your command files (more about command files later). You can use the DRAW command as often as you like to be sure that the plotting commands you type are drawn and not erased.

TEXT

The Graphics Machine also lets you to add lines of text to your screen images.

Command Syntax: TEXT

```
> TEXT x1,y1
ENTER STRING your text
```

x1,y1-- The point at which to place beginning of text, relative to the graphics mode zero screen.

your text-- After the "ENTER STRING" prompt, type any text you wish to place on the screen. Your text must be less than 40 characters long.

The location of your text is not relative to the Graphics Mode eight screen, but to the Graphics Mode zero screen (ATARI text mode). This means that, when placing text on the screen, you must think of its position in terms of different coordinate systems. The graphics mode zero screen measures 40 by 23 units, but there are several factors which determine the Graphics Machine's placement of text.

In actuality, the text placement is determined by a 40 by 23 coordinate system with pixel control when the x-coordinate exceeds 40 and is less than 255. A pixel is the smallest point the computer's hardware can plot on the screen. X-coordinates greater than 40 move the text one pixel down on the screen, but your text remains relative to the Graphics Mode Zero y-coordinate. Thus you may control exactly where your text is placed on the screen. The 40 by 23 system is relative to the size of the characters, while the 255 by 23 system corresponds to the physical capabilities of the screen.

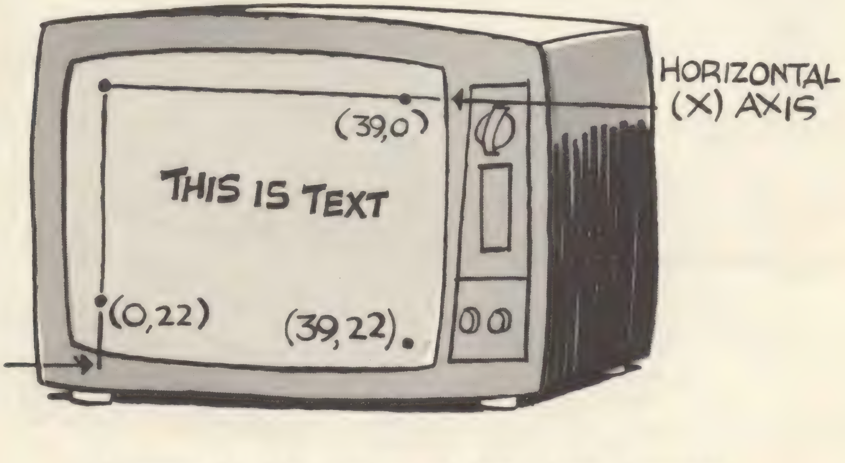


Figure 2: The Graphics Mode Zero Screen

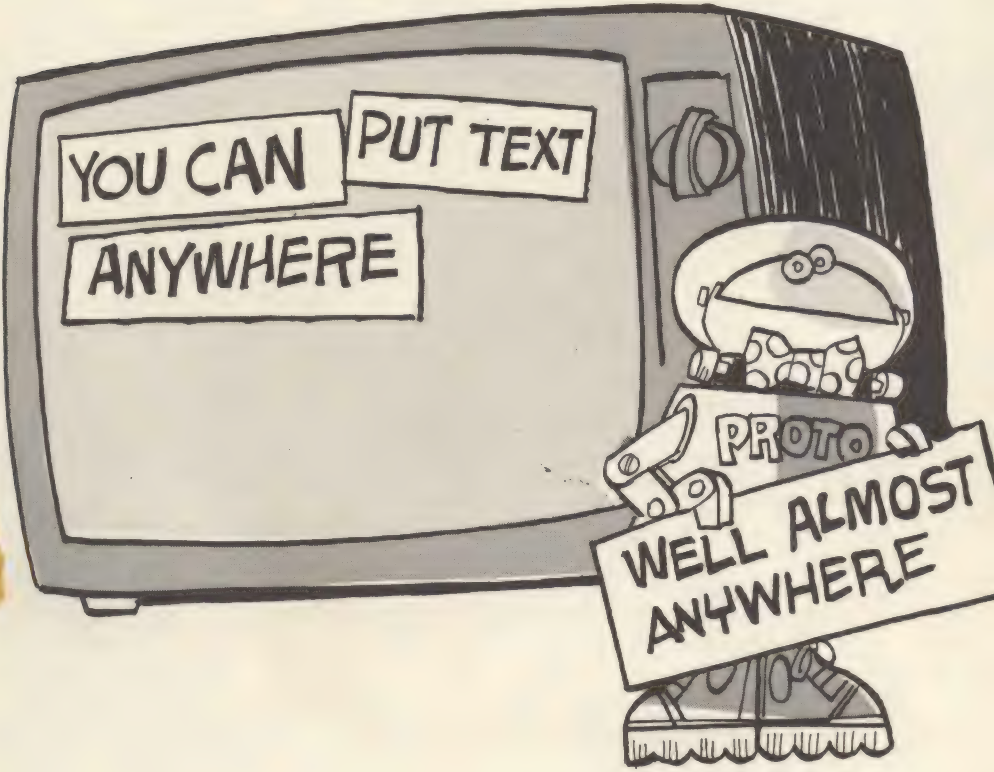
After you tell the Graphics Machine where to place your text, you must tell it what that text is. You enter your text string after the "ENTER STRING" prompt. The Graphics Machine allows a string of up to 40 characters as a TEXT command parameter. If the length of your text exceeds the number of text positions on the current line, it will wrap around the screen and appear one pixel lower.

To see some examples of TEXT placement, type:

```
> FILER 0
filename==>? TEXT.DEM
```

The TEXT demo shows you the following commands:

```
> TEXT 1,10
ENTER STRING HELLO
> TEXT 41,10
ENTER STRING HELLO
> TEXT 81,10
ENTER STRING HELLO
> TEXT 1,12
ENTER STRING HELLO
```



COLOR COMMANDS

There are two ways to add colors to your graphics display; you can change the screen color with the SETCOLOR command, and you can fill retangular areas with white, red, blue, and black using the FILL command.

The exact colors you see on your screen always depend on how you have the color settings on your TV.

SETCOLOR

The SETCOLOR command is the same as the ATARI BASIC Setcolor command (see page 50 in the ATARI BASIC Manual). With this command, you can change the color of the background and border of the screen. You must specify the color register, luminance, and hue.

Command Syntax: SETCOLOR

```
> SETCOLOR color register, luminance, hue
```

```
Color register--  Determines colored area of screen.
                  Must be either 2 or 4.
hue--             Determines color. Ranges form 0 to 15.
luminance--       Determines brightness of color.
                  Ranges from 0 to 14, even numbers only.
                  A luminance of 14 is almost white.
```

The register you choose determines where color will appear on the screen. If you specify color register 2, the working area oт your screen is the color you specify with the hue and luminance parameters. If you specify color register 4, the border of your screen is colored, and the main working area remains black.

The hue determines the color of the screen. Hues range from 0 to 15. For a list of hue colors and their coıresponding numbers, see page 50 of the ATARI BASIC Manual.

The luminance determines the brightness of that color. It must be an even number between 0 and 14. The larger the number, the brighter your screen; a luminance of 14 is almost pure white.

FILL

The FILL command fills rectangular areas of the screen with white, blue, or red.

Command Syntax: FILL

```
> FILL x1,y1,x2,y2,skip
```

x1,y1-- The coordinates of the upper left-hand corner of the rectangular area to fill.
x2,y2-- The coordinates of the lower right-hand corner.
skip-- The number of graphics mode eight lines to skip horizontally.

Think of your screen as an array of microscopic colored light bulbs. The odd numbered columns of the array are all red light bulbs, and the even numbered columns are blue light bulbs. Figure 3 shows you a "magnified" section of this imaginary screen.

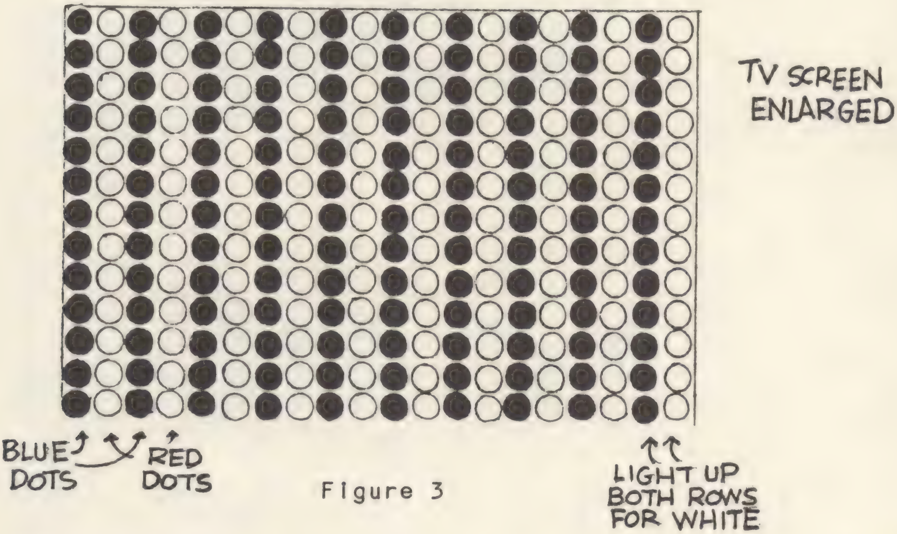


Figure 3

When you turn on all of the light bulbs in a given area, the screen looks white, If you start with an odd-numbered column and only turn on the lights in every other column, you are turning on only red lights, and the screen appears red. If you start at an even column and turn on the lights in every other column, you are turning on only blue lights, and the screen looks blue. If you do not turn on any of the lights, the screen remains black. In fact, the physical properties of a color television are much the same as our imaginary array of lights. The Graphics Machine uses these physical capabilities to put color on your screen (this process is called artifacting.)

In order to fill a certain area, you must tell the Graphics Machine the dimensions of that area, and the number of "columns" to skip (the skip value).

The dimensions are the same as those required by the BOX command: the coordinates of the upper left-hand corner (x1,y1), and the coordinates of the lower right-hand corner (x2,y2).

The number of columns that you skip determines the color of your box. If you enter a skip value of 1, the Graphics Machine fills every column; your box is white. If your box has a value of x1 which is an even number, and you enter a skip value of 2, the Graphics Machine "lights" every other column; your box is blue. If you start with an odd value for x1 and skip 2 columns, your box is red.

To see how the FILL command works, type:

```
>FILER 0
filename==>?FILL1.DEM
```

The FILL1 demo uses the following command to show you a white box, a blue box, and a red box.

```
> FILL 10,10,50,50,1
> FILL 60,20,100,60,2
> FILL 105,30,145,70,2
```

If you enter a skip value greater than 2, your box will appear striped. For example, a skip value of 5 lights, every fifth column, your box will consists of red and blue stripes. Try entering this command:

```
> FILL 10,80,80,150,5
```

The FILL command also works in ERASE mode. Instead of turning on the lights in the imaginary array, you are turning them off. If you fill a rectangle with white in DRAW mode, then enter ERASE mode and fill the same rectangle with a skip value of 2, you erase every other column,and the rectangle turns red or blue, depending on the x1-coordinate of your FILL command. To see an example of how to use FILL with ERASE mode, type the following:

```
> FILER 0
filename==>?FILL2.DEM
```

The FILL2 demo shows how to make a white box red or blue with the following commands:

```
> FILL 10,10,100,100,1
     This command draws a white box.
> ERASE 0
> FILL 10,10,50,50,2
     This command erases blue "columns".
> FILL 11,10,100,100,2
     This command erases red "columns".
```

SCREEN COMMANDS

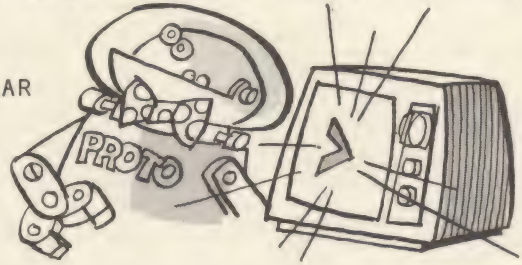
Once you have created a graphic display on your screen, you may want to change the status of the screen. For example, you may want to erase a whole screen, or see the images hidden by the command window. The Graphics Machine has several commands to change the status of the whole screen.

CLEAR

The CLEAR command clears the entire graphics screen. Be careful, once you clear a screen, there is no way to recall the data that was on it.

Command Syntax: CLEAR

```
> CLEAR 0
```



FULLSCREEN

The FULLSCREEN command shows you the full graphics screen, without the command window, for the number of seconds you specify.

Command Syntax: FULLSCREEN

```
> FULLSCREEN number of seconds
```

number of seconds-- Determines the number of seconds that the Graphics Machine shows the full screen.

Press any key to return to normal command mode.

The Graphics Machine returns to normal command mode if you press any key. For example, you can tell the Graphics Machine to show you the full screen for 999 seconds, and then press any key to return to command mode before 999 seconds have passed.

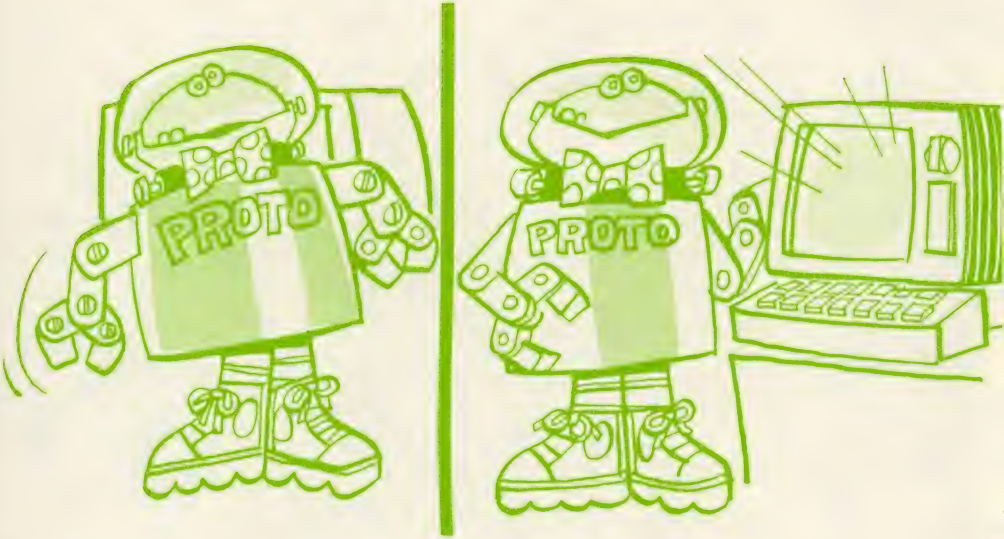
HIDESCREEN

The HIDESCREEN command "turns the screen off" so that the Graphics Machine can process commands faster.

```
Command Screen: HIDESCREEN

    > HIDESCREEN 0
```

The screen is blank while you enter command either from the keyboard or from a disk file. To turn the screen back on, use the SHOWSCREEN command.



SHOWSCREEN

The SHOWSCREEN command turns the screen back on after a HIDESCREEN command. This is the only way to see your graphic images again after using the HIDESCREEN command.

```
Command Syntax: SHOWSCREEN

    > SHOWSCREEN 0
```

DISK FILES

When you finish creating a graphic design on your screen, you will probably want to save it. The Graphic Machine lets you save screen images on disks in two ways: in screen data files, and in command files.

DIR

The DIR command displays the directory of the disk in the drive you specify.

```
    Command Syntax: DIR

          > DIR drive number

 drive number-- The number of the disk drive of which
                you want a directory.
```

If you enter a nonexistent drive number, you must wait for the Graphics Machine to complete its search for that drive and then press START to continue.

Screen Data Files

SAVESCREEN

The SAVESCREEN command lets you save the images you see on tne screen in a disk file.

```
     Command Syntax:SAVESCREEN

          > SAVESCREEN 0
          filename==>? your file

  your file-- Any file name of your choice up to eight
              characters long. The Graphics Machine adds
              the ".SCR" extender.
```

The file contains the exact screen data, not the commands you have used to create that data. The Graphics Machine prompts you for the name of your data file. File names can be any word less than or equal to eight characters. The Graphics Machine adds ".SCR" to the end of your file name in order to designate it as a screen data file.

A screen data file requires 62 disk sectors; use the DIR command to determine whether the disk has enough room before saving a data file.

GETSCREEN

The GETSCREEN command retrieves a screen data file from the disk; it puts the images you saved with the SAVESCREEN command back onto the screen.

Command Syntax: GETSCREEN

```
> GETSCREEN 0
filename==>? your file
```

your file-- Any screen data file you have created with the SAVESCREEN command. Do not type ".SCR".

COMMAND FILES

The Graphics Machine also lets you create disk files containing Graphics Machine commands. When it executes such a file, it executes each command one after the other. You actually see the commands in the command window as the Graphics Machine executes them. The demonstration examples you looked at were all examples of command files.

Creating Command Files: The &

The Graphics Machine lets you create a command file as you enter commands on the screen.

Command Syntax: &

```
> & any command, command parameters
```

any command-- Any Graphics Machine command that does not prompt for additional information.

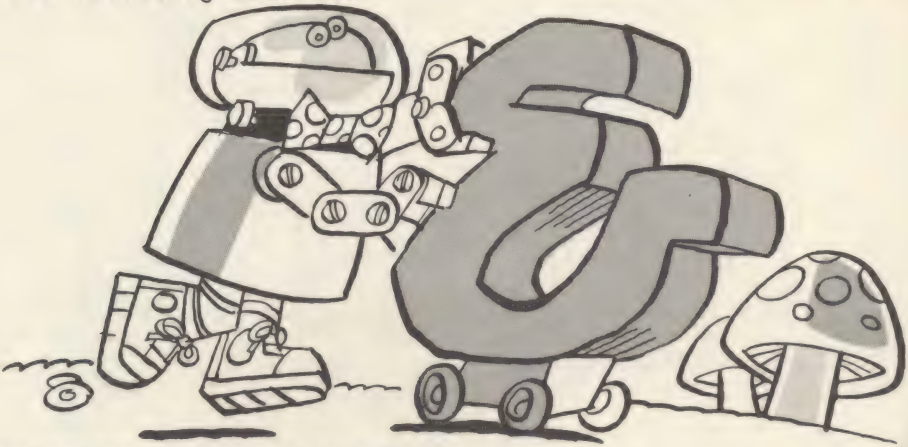
command parameters-- The parameters corresponding to your command.

Repeat this format with all commands you want to put into the file.

If you type the character "&" (an ampersand--shift 6 on your keyboard) before a command, the Graphics Machine opens a special command file called "OUT". The Graphics Machine places that command and any others that follow it and are also preceded by an "&" in the disk file OUT.

You cannot send commands which prompt for additional information (TEXT, SAVESCREEN, GETSCREEN) to the file OUT.

There is an example showing how to create a command file in the following section.



Closing the OUT file: CLOSEOUT

The CLOSEOuT command closes the OUT file which you created with the commands preceded by the "&".

Command Syntax: CLOSEOUT

```
> CLOSEOUT 0
```

For example, create the following command file using the & ana CLOSEOUT commands.

```
> &BOX 0,0,319,191
> &BOX 20,20,175,150
> &BOX 40,40,100,100
> &FULLSCREEN 10
> CLOSEOuT 0
```

The Graphics Machine names this file "OUT". You can create only one OUT file each session; you must re-boot the Graphics Machine if you wish to create a new OUT file. When you create a new OUT file, however, you destroy the previous OUT file. To prevent this, simply change the name of the file from OUT to any other legal file name using the ATARI Disk Operating System.

Retrieving a Command File: the FILER command

The FILER command tells the Graphics Machine to get command files from the disk. It prompts you for a file name.

Command Syntax: FILER

```
> FILER 0
filename==>? any command file
```

any command file-- Any command file on disk.

To retrieve the file named OUT which you just previously created, type:

```
> FILER 0
filename==>? OUT
```

The Graphics Machine executes all commands input from the file until it reaches the end of the file. You see any errors or bad commands in the disk file. You must have the file DERER in disk drive one, if you have used the DISKERRORS command for long error messages.

Advanced File Creation

If you have an ASSEMBLER Cartridge, you can create command files with the editor and then send them to the disk. These files can contain all commands, including those which require additional input. Use the FILER command to retrieve these files from the disk.

Format for ASSEMBLER/EDITOR file creation

line-number COMMAND command- parameters

line-number--	The consecutive number of the current line.
COMMAND--	The Graphics Machine command you want to put into the file.
command-parameter--	The parameters corresponding to your command.

Each line must be numbered as you number a BASIC program, and there must be a space after the line number for the Graphics Machine to interpret the command correctly. This syntax is identical to the Assembler/Editor Cartridge. Using the NUM command, you can create files easily and efficiently (for more details see the Assembler/Editor Manual).

A Final Word

We are always in the process of updating our software. Please let us know if you find any strange bugs, or if you have any difficulties using the Graphics Machine.

Any defective disks or tape should be returned to:
Educational Software, Inc.
4565 Cherryvale Ave.
Soquel, CA 95073
(408) 476-4901

The master disk is not copy protected. Please make personal copies only.

HAVE FUN!

COMMAND INDEX AND SYNTAX SUMMARY

Remember, 0 is a dummy parameter; it must accompany designated commands.

BOX

```
> BOX x1,y1,x2,y2
```

x1-- x-coordinate of the upper left-hand corner.
y1-- y-coordinate of the upper left-hand corner.
x2-- x-coordinate of the lower right-hand corner.
y2-- y-coordinate of the lower right-hand corner.

Command Description: page 6

CIRCLE

```
> CIRCLE x,y,radius,roundness
```

x,y-- The coordinate of the center point of your circle.

radius-- The radius, measured in horizontal units from the center.

roundness-- The roundness of a perfectly round circle is one. Numbers greater than one elongate the circle in the horizontal direction; numbers less than one and greater than zero elongate the circle in a vertical direction.

Command Description: page 7

CLEAR

```
> CLEAR
```

Command Description: page 15

& Creating the OUT file

```
> &any command, command parameters
```

any command-- Any Graphics Machine command that does not prompt for additional information.

command parameters-- The parameters corresponding to your command.

Repeat this format with all commands
you wish to put in a file.

Command Description: page 19

CLOSEOUT

```
> CLOSEOUT 0
```

Command Description: page 19

DIR

```
> DIR drive number
```

drive number-- The number of the disk drive of which you want a directory.

Command Description: page 17

DISKERRORS

```
> DISKERRORS 0
```

Command Description: page 4

DRAW

```
> DRAW 0
```

Command Description: page 9

ERASE

```
> ERASE 0
```

Command Description: page 9

ERROROFF

```
> ERROROFF 0
```

Command Description: page 4

FILER

```
> FILER 0
filename==>? any command file
```

any command file-- Any command file on disk.

Command Description: page 20

FILL

```
> FILL x1,y1,x2,y2,skip
```

x1,y1-- The coordinates of the upper left-hand corner of the rectangular area to fill.
x2,y2-- The coordinates of the lower right-hand corner.
skip-- The number of Graphics Mode Eight lines to skip horizontally.

Command Description: page 13

FULLSCREEN

```
> FULLSCREEN number of seconds
```

number of seconds-- Determines the number of seconds that the Graphics Machine shows the full screen.

Press any key to return to normal command mode.

Command Description: page 15

GETSCREEN

```
> GETSCREEN 0
filename==>? your file
```

your file-- Any screen data file you have created with the SAVESCREEN command.

Command Description: page 18

HELP

> Type the word HELP anywhere on command line. Press RETURN.

Command Description: page 4

HIDESCREEN

```
>HIDESCREEN 0
```

Command Description: page 16

LINE

```
> LINE x1,y1,x2,y2
```

x1-- The x-coordinate of the first endpoint.
y1-- The y-coordinate of the first endpoint.
x2-- The x-coordinate of the second endpoint.
y2-- The y-coordinate of the second endpoint.

Command Description: page 5

POLYGON

```
> POLYGON x,y,radius,number of sides,angle of
  rotation
```

x,y-- The coordinates of the center of the polygon.

radius-- The radius measured in horizontal units from the center.

number of sides-- The number of sides of the polygon.

angle of rotation-- The measure in degrees of the rotation of the starting point of the plot. An angle of zero degrees starts the plot at the point. (x1+radius,y1)

Command Description: page 8

SAVESCREEN

```
> SAVESCREEN 0
filename==>? your file
```

your file-- Any file name of your choice up to eight characters long. The Graphics Machine adds the ".SCR" extender.

Command Description: page 17

SETCOLOR

```
> SETCOLOR color register, luminance, hue
```

color register-- Determines colored area of screen. Must be either 2 or 4

luminance-- Determines brightness of color. Ranges from 0 to 14, even numbers only. A luminance of 14 is almost white.

hue-- Determines color. Ranges from 0 to 15.

Command Description: page 12

SHOWSCREEN

```
> SHOWSCREEN 0
```

Command Description: page 16

TEXT

```
> TEXT x,y
ENTER STRING your text
```

x,y-- The point at which to place beginning of text.

your text-- After the "ENTER STRING" prompt, type any text you wish to place on the screen. Your text must be less than 40 characters long.

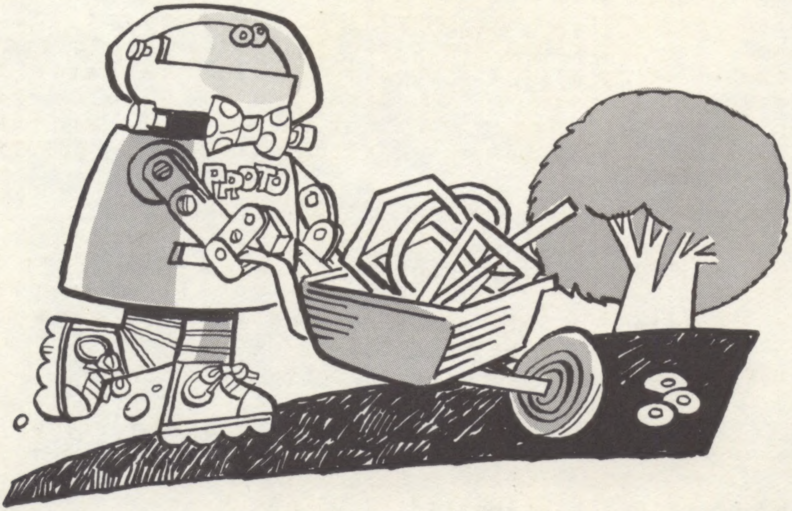
Command Description: page 10

TRIANGLE

```
> TRIANGLE x1,y1,x2,y2,x3,y3
```

x1,y1-- The coordinates of the first point.
x2,y2-- The coordinates of the second point.
x3,y3-- The coordinates of the third point.

Command Description: page 6

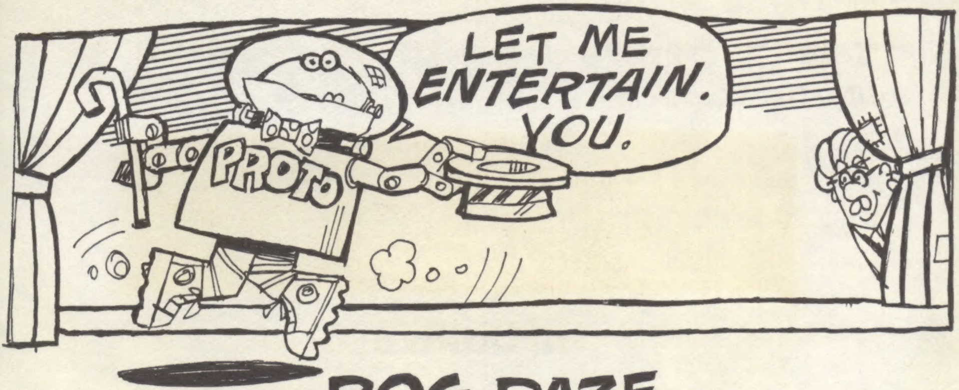


ALSO FROM EDUCATIONAL SOFTWARE

PROTOTYPE'S ADVENTURES

Ages 4 to 8

by Stan Gilbert. A delightful collection of 8 NON-VIOLENT and NON-COMPETITIVE games written especially for young children. Professor Von Chip's robot, Proto Type, will guide your child through activities such as catching free MARSHMALLOWS from friendly aliens, electronic COLORING, JUMP ROPE, and a fun game called ALPHA BLOCKS. Proto even shows them where he lives and works. The storybook, shaped like Proto Type, has cartoons for your child to color that don't need a computer to be fun. 16K-Tape/24K-Disk -- $19.95



DOG DAZE

by Gray Chang. A fast action arcade style game where the players are puppies fighting for control of the local fire hydrants. Each pooch races to claim the hydrants either by touching it first or by shooting his bone at it. And doglike, if a dog comes too close to another dog's hydrant, he's compelled to stop and sniff, losing valuable time. Each dog must be careful of cars or the game could end rather suddenly! Great fun for all ages! Written in efficient machine language. 8K-Tape/24K-Disk -- $16.95

SPACE GAMES

by Randy Massey. Three exciting games for one very affordable price. With all the fun you'll have playing ALIEN, SURVIVE, and ROBOT ATTACK you can't lose even if you get turned into space dust! 16K-Tape/32K-Disk -- $24.95 for all three.

**IN THE UNLIKELY EVENT
YOU RECEIVE A DEFECTIVE PRODUCT
RETURN IT TO US FOR A
PROMPT REPLACEMENT OR REFUND**

GRAPHICS MACHINE

Turn your ATARI computer into a powerful graphics machine! You can type in simple commands such as LINE, CIRCLE, and POLYGON to get professional looking high resolution screens which you can SAVE on disk in only five seconds! Retrieval is just as fast. All graphics are in mode 8 with three colors and you can add text to any screen at will.

Also included is the ability to save each command as you enter them so you can recreate the sequence later on. Great fun and a good usable tool besides!

REQUIRES:

48 Disk only

Educational
Software inc.

4565 Cherryvale Ave
Soquel, Ca. 95073
(408) 476-4901